Linear Associative Memory Dynamics and CCSLM

A technical note for time-pressed researchers

1. The central observation

A useful way to understand a linear mapping is as a **linear associative memory**, rather than merely as a parameter matrix.

Under this interpretation, initialization is already an associative-memory state. Training does not necessarily replace that state. It modifies it.

This gives a different perspective on several familiar phenomena:

- initialization can have surprisingly long-lasting effects;
- continued SGD does not necessarily erase earlier information;
- weight decay can cause substantial changes even when training error is already zero;
- over-capacity memories exhibit statistical recall rather than complete forgetting;
- learning rate controls the rate at which older associations are corrupted;
- training history can remain encoded in the final mapping.

These observations are particularly relevant to **Concatenated Context-Selected Linear Mappings (CCSLM)**, where a system is viewed as a sequence of context-selected linear mappings.

---

2. Capacity changes the meaning of optimization

Consider a weighted sum trained to associate input vectors with scalar outputs.

If the number of independent associations is below the effective capacity of the weighted sum, there will generally be many weight vectors that satisfy the training constraints.

Consequently, after SGD has found one exact solution, removing one training example does not necessarily cause any movement at all.

The existing weight vector remains a valid solution to the reduced problem.

This is an elementary but important point:

> **Continued training does not imply continued movement. If the current associative state already satisfies the active constraints, there may be no optimization pressure to change it.**

The situation changes if weight decay is introduced.

After removing an association, weight decay can move the weight vector toward another member of the solution family having a smaller norm. The mapping can therefore change substantially while continuing to satisfy all remaining associations.

This is better understood as **movement within a solution manifold** than as ordinary forgetting.

---

3. Weight decay as associative forgetting

The usual interpretation of weight decay is that it discourages large weights.

For a linear associative memory, a more revealing interpretation is:

> **Weight decay imposes a systematic drift toward lower-norm associative states.**

This can have very different consequences depending on capacity.

Under capacity

There may be many exact solutions. Weight decay can therefore move the mapping from one exact solution to another, reducing its norm without necessarily degrading recall of the remaining associations.

The change can involve a substantial change in direction, not merely uniform shrinkage.

At or near capacity

The constraints leave less freedom for such movement. Weight decay increasingly competes with the requirement to preserve associations.

Over capacity

The mapping cannot satisfy all associations simultaneously. Weight decay is then one component of a larger compromise between fitting associations and controlling the magnitude of the associative state.

This suggests that regularization choices for CCSLM should be considered in terms of **associative-memory dynamics**, rather than imported uncritically from conventional neural-network practice.

---

4. The weight-vector path-walk view

For sequential SGD training of a linear predictor, an update can be written schematically as

$$w_{t+1} = w_t + \eta x_t (y_t - w_t^T x_t).$$

The important point is conceptual rather than the particular update rule.

Each new association produces a perturbation of the existing associative state.

Thus the weight vector follows a trajectory through parameter space:

$$w_0 \to w_1 \to w_2 \to \cdots \to w_t.$$

Every update potentially changes the recall behaviour of previously learned associations.

For an over-capacity memory, the trajectory can be viewed approximately as a **stochastic walk through associative state space**:

- the current example pulls the weight vector toward better recall of that example;
- the update perturbs the distributed representation of previous associations;
- the perturbation introduces noise into older recalls;
- repeated updates gradually accumulate interference.

This provides an intuitive mechanism for recency effects.

A recently presented association has been reinforced by relatively few subsequent updates. An old association has experienced many subsequent perturbations.

Thus:

> **Recency can emerge naturally from accumulated associative interference, without requiring an explicit memory-management mechanism.**

Importantly, deterioration of an old association does not mean that its information has suddenly been deleted. It can instead be progressively corrupted by subsequent movement of the shared weight vector.

---

5. Over-capacity does not mean immediate forgetting

An over-capacity weighted sum cannot perfectly satisfy all associations, but it can still exhibit substantial **statistical recall**.

This is an important distinction.

The old association can remain statistically represented in the weight vector even after many subsequent updates. Its recall becomes noisier rather than immediately disappearing.

A useful qualitative picture is:

new association → perturbation → interference with old associations → further perturbation → gradual accumulation of recall noise.

Learning rate then becomes a memory-timescale parameter.

A smaller learning rate produces smaller perturbations per presentation and can therefore permit much longer persistence of old associations. A larger learning rate produces stronger interference and shorter effective memory.

Repeated presentation of an old association can counteract this accumulated interference.

This is closely related to continual-learning and catastrophic-forgetting phenomena, but the associative-memory interpretation emphasizes the underlying mechanism:

> **The weight vector is itself the distributed memory, and learning is a continuous modification of that memory.**

---

6. Initialization is therefore part of the memory state

If a random linear mapping is already an associative memory, initialization should not be regarded solely as a numerical device for starting optimization.

The initial matrix contains a distributed collection of random associations.

Training modifies this state.

There is therefore no general reason to expect the initial associations to disappear merely because a large number of optimization steps have occurred.

In some regimes, the training trajectory can retain substantial statistical information about its starting point.

This makes initialization a potentially fundamental CCSLM design variable.

Possible initialization regimes include:

- very small-magnitude random weights;
- very large-magnitude random weights;
- orthogonal matrices;
- volume-preserving / determinant-one mappings;
- other deliberately structured initial associative states.

These are not simply different ways of getting SGD started. They can correspond to **different initial associative-memory conditions**.

---

7. Connection with Saxe and deep linear networks

Saxe and collaborators' work on deep linear networks is particularly relevant here.

Their analytical treatment showed that initialization and weight scale can have major effects on learning dynamics. Orthogonal initialization is especially interesting because it can preserve signal geometry and avoid immediately imposing severe contraction or expansion on different directions.

From a CCSLM perspective, this suggests an additional interpretation:

> **Orthogonal initialization provides a relatively neutral initial associative geometry from which regional linear experts can specialize.**

This is potentially important for deep chains because early rank loss, poor conditioning, or strong directional contraction can reduce the information available to subsequent mappings.

Saxe's work therefore provides useful conceptual and practical precedents even though it does not make the CCSLM identification of ReLU networks.

---

8. Why ReLU and Atlas-LSH CCSLM may behave differently

The persistence of initialization and the effects of regularization need not be identical across architectures.

In a conventional ReLU network, the effective linear mapping is coupled to local activation patterns. The context determining which linear regime is active is itself generated locally by the network.

An Atlas-LSH CCSLM has a different structure.

A global input vector is used to generate an LSH context, and that context selects the regional linear expert used at a computational stage.

Consequently, the local expert receives a much more explicit contextual partition of the associative workload.

This can substantially change its effective capacity and interference pattern.

The hypothesis suggested by practical experiments is that the regional experts in an Atlas-LSH system may preserve their associative state more readily than the corresponding mappings in conventional ReLU networks.

This should be tested systematically rather than assumed.

---

9. CCSLM as factorized associative memory

A useful high-level interpretation of CCSLM is:

> **CCSLM factorizes a large associative-memory problem into a sequence of context-selected local associative memories.**

Instead of requiring one mapping to represent every association, context distributes associations among regional experts.

An individual expert may therefore operate in an over-capacity regime even when the complete CCSLM has very large effective capacity.

This changes the role of interference.

If similar inputs tend to produce similar LSH contexts, related associations will tend to train the same or nearby regional experts. LSH therefore does more than provide routing: it controls the **distribution of associative interference**.

This suggests that context design and associative-memory capacity are closely connected.

---

10. Training CCSLM

There is unlikely to be a single universally optimal training procedure.

Relevant choices include:

- initialization scale;
- orthogonal versus unconstrained initialization;
- learning rate;
- learning-rate schedule;
- weight decay;
- explicit norm constraints;
- rank constraints;
- pseudoinverse or least-squares initialization;
- SGD or other iterative optimization;
- degree of coupling between related regional experts;
- context overlap;
- expert size;
- number of computational stages.

The appropriate choices are likely to depend strongly on the particular CCSLM problem.

For example, if persistence of learned associations is desirable, aggressive weight decay may be counterproductive. If uncontrolled accumulation of obsolete associations is undesirable, some controlled forgetting mechanism may instead be useful.

Likewise, orthogonal initialization may be particularly attractive where preservation of observability through a deep chain is important.

---

11. Experimental programme

Several simple experiments could clarify the dynamics considerably.

Initialization persistence

Train otherwise identical mappings from:

- small random weights;
- large random weights;
- orthogonal weights;
- determinant-one random mappings.

After very long training, measure similarity between the initial and final mappings.

Capacity sweep

Train linear associative memories at:

- well below capacity;
- near capacity;
- above capacity;
- substantially above capacity.

Measure both exact recall and statistical recall.

Sequential interference

Present associations sequentially and measure recall as a function of their age.

This directly tests the predicted relationship between learning rate and associative lifetime.

Removal experiment

Train below capacity, remove one association, and continue training:

1. without weight decay;
2. with weak weight decay;
3. with stronger weight decay.

Measure the movement of the weight vector and its norm.

CCSLM expert dynamics

Perform the same measurements separately for regional experts in an Atlas-LSH CCSLM.

In particular, measure:

- expert utilization;
- expert capacity;
- associative interference;
- initialization persistence;
- mapping similarity over time;
- relationship between LSH collision structure and interference.

---

12. A broader interpretation

The conventional neural-network description treats a weight matrix primarily as a set of parameters optimized to minimize a loss.

The associative-memory perspective adds a temporal dimension:

> **A trained linear mapping is a state with history.**

Its present behaviour depends not only on the current training constraints but potentially on:

- its initialization;
- the sequence of associations it has encountered;
- their presentation frequencies;
- learning rate;
- regularization;
- capacity;
- and the trajectory by which the current state was reached.

This provides a natural explanation for why apparently elementary experiments can produce unexpectedly persistent effects.

For CCSLM, this may be particularly important because the system is explicitly constructed from linear mappings. Understanding the **dynamics of linear associative memories** may therefore be more informative than treating those mappings as generic neural-network weights.

Summary

The useful conceptual chain is:

linear mapping → associative memory → capacity → solution manifolds → SGD trajectory → interference → statistical recall → persistence → contextual factorization → CCSLM.

Saxe's deep-linear work provides an important part of this picture, particularly concerning initialization and learning dynamics. The CCSLM perspective extends the emphasis from *how a deep linear network learns a mapping* to *how a collection of context-selected linear associative memories evolves and retains history*.

A particularly promising question is therefore:

> **What are the characteristic dynamical laws of a linear associative memory under sequential learning, and how should CCSLM initialization and training be designed to exploit them?**